UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/533,578	02/03/2006	Noel William Lovisa	340512-900307	4468

73230   7590   01/27/2011
DLA PIPER US LLP
1999 AVENUE OF THE STARS
SUITE 400
LOS ANGELES, CA 90067-6023

EXAMINER
VU, TUAN A

ART UNIT	PAPER NUMBER
2193	

NOTIFICATION DATE	DELIVERY MODE
01/27/2011	ELECTRONIC

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Notice of the Office communication was sent electronically on above-indicated "Notification Date" to the following e-mail address(es):

laura.rivero@dlapiper.com
laccusptomail@dlapiper.com
cliff.keyner@dlapiper.com

	Application No.	Applicant(s)
Office Action Summary	10/533,578	LOVISA ET AL.
	Examiner	Art Unit	
	TUAN A. VU	2193	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE *3* MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.
- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *22 November 2010*.
2a)☒ This action is **FINAL**.   2b)☐ This action is non-final.
3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-34,37-43,46-48,51-59 and 62-69* is/are pending in the application.
    4a) Of the above claim(s) _____ is/are withdrawn from consideration.
5)☐ Claim(s) _____ is/are allowed.
6)☒ Claim(s) *1-34, 37-43, 46-48, 51-59, 62-69* is/are rejected.
7)☐ Claim(s) _____ is/are objected to.
8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.
10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.
    Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
    Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
    a)☐ All   b)☐ Some * c)☐ None of:
      1.☐ Certified copies of the priority documents have been received.
      2.☐ Certified copies of the priority documents have been received in Application No. _____.
      3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).
    * See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1)☐ Notice of References Cited (PTO-892)
2)☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3)☐ Information Disclosure Statement(s) (PTO/SB/08)
    Paper No(s)/Mail Date _____.
4)☐ Interview Summary (PTO-413)
    Paper No(s)/Mail Date. _____.
5)☐ Notice of Informal Patent Application
6)☐ Other: _____.

DETAILED ACTION

1.   This action is responsive to the Applicant's response filed 11/22/10.

As indicated in Applicant's response, claims 1, 4, 7, 15, 18, 20-22, 24, 29-31, 37-39, 41,

43, 46, 48, 51, 53, 55-59 have been amended, and claims 66-69 added. Claims 1-34, 37-43, 46-

48, 51-59, 62-69 are pending in the office action.

Double Patenting

2.   The nonstatutory double patenting rejection is based on a judicially created doctrine
grounded in public policy (a policy reflected in the statute) so as to prevent the unjustified or
improper timewise extension of the "right to exclude" granted by a patent and to prevent possible
harassment by multiple assignees. See In re Goodman, 11 F.3d 1046, 29 USPQ2d 2010 (Fed.
Cir. 1993); In re Longi, 759 F.2d 887, 225 USPQ 645 (Fed. Cir. 1985); In re Van Ornum, 686
F.2d 937, 214 USPQ 761 (CCPA 1982); In re Vogel, 422 F.2d 438, 164 USPQ 619 (CCPA
1970);and, In re Thorington, 418 F.2d 528, 163 USPQ 644 (CCPA 1969).
        A timely filed terminal disclaimer in compliance with 37 CFR 1.321(c) may be used to
overcome an actual or provisional rejection based on a nonstatutory double patenting ground
provided the conflicting application or patent is shown to be commonly owned with this
application. See 37 CFR 1.130(b).
        Effective January 1, 1994, a registered attorney or agent of record may sign a terminal
disclaimer. A terminal disclaimer signed by the assignee must fully comply with 37
CFR 3.73(b).

3.   Claims 14, 10, 51-52, 38 are provisionally rejected under the judicially created doctrine

of obviousness-type double patenting as being unpatentable over claims 11, 17, 6, 20 of

copending Application No. 10,533, 577 (hereinafter '577).

Although the conflicting claims are not identical, they are not patentably distinct from

each other because of the following observations. Following are but a few examples as to how

the certain claims from the instant invention and from the above copending application are

conflicting with each other.

As per instant claim 14, '577 claim 11 recites user to obtain a desired service using a

processing system including determining a combination of selected components from the user,

each component associated with a service and provided by different entities, implementing the

component by a service request to each entity to request the service to be performed. Hence '577

has recited a obvious language variant of instant claim 1 reciting of implementing a component

for code generating, component combinations with requirements to allow desired functionality to

be achieved, component servers, each to correspond to said component in the combination,

component server to perform the service. Further '577 claim 11 recites 'combination … in

accordance with input received from the user … combination defining a sequence of service

portions and interconnections between components defining transfer of data' ('577 claim 1),

hence this would be a obvious variant of instant claim 9 implemented by one or more processing

systems. Further, '577 claim 11 recites end/base station to generate a graphical representation of

the selected components, manipulate the representation in response to user commands, transfer

the graphical representation to the end station, which is considered a same variant of the

'graphical representation' recited in instant claim 14.

  But '577 does not explicitly recite each server to perform data manipulation service in

accordance with defined series, servers performing service by interacting with data sequence,

including interacting among servers and causing further implementation with obtaining of results

from the series of data manipulation, the resultant data sequence being the computer code,

providing the computer code to a processing system, causing this to perform the desired

functionality.

  However, '577 claim 11 recites base station manipulating the graphical representation in

response to the received user commands, transferring the representation to the end station, to

implement the combined components in accordance to the graphical representation and (in '577

claim 1) different components provided by different entities … input received from the user …

combination defining a sequence of service portions and interconnections between components

defining transfer of data". In view of '577 recited of 'service request … transferred to each

entity … respective to a portion … causing the desired service to be performed' ('577 claim 1) it

is recognized that '577 has disclosed a obvious variant of: manipulation service being a request

portion to be performed via receiving first system user's request, of interacting among entities via

transfer of portion to be performed, so that the combined resultant from each separate

'manipulation' service portion by each said entity or base station results in the target component

which when transmitted to the first end station would achieve the implementation of the user

desired functionality (i.e. code implementation to execute that functionality). That is, based on

the interconnections define in the defined sequence and the transfer between entities based

thereon, '577 claim 11 has disclosed a obvious variant of 'manipulation service' being

distributed among entities or servers so to include interaction among these entities, including

transfer and obtaining of resulting data from each portion being performed by each "respective"

serving entity.

    As per instant claim 10, '577 claim 17 discloses receiving information to be

manipulated at ports, performing manipulation and providing the manipulated information at

ports; hence has disclosed an obvious variant of 'manipulation service' performed by an server

entity in response to user definition of component combination or user inputs and indication of

ports of each said component including interconnect ones of the ports in response to the user

commands in view of the user's role as set forth above (see '577 claim 1).

As per instant claim 51, this claim includes the same language as instant claim 1, hence in view of the teaching of '577 claim 11 with manipulation of data by the server receiving user commands, user's defined interconnected components to describe the sequence of 'service portions' to be performed by different entities and transfer between entities, '577 claim 11 would represent a obvious language representing instant claim 51.

As per instant claim 52, '577 claim 6 recites base station storing of port specifications and providing port information to the end station for user to select, hence has disclosed a obvious variant of instant claim 52.

As per instant claim 38, '577 claim 20 also recites 'providing performance information to the user for user's selection, including entity, geographical location, duration, cost associated with implementing, rating, but does not recite 'details' by one or more processing systems coupled to end stations, said systems to provide details to users, details of components or based on component specifications from each entity that perform respective manipulation service. Based on the provision of information and a cost corresponding to implementation by each such entity, the concept of storing details by a server entity is evident, as in '577 claim 20. Based on the transfer between entities to achieve the overall functionality effected as portion of service each performed by a respective entity, the overall functionality implemented from sequence of components defined by the user to request inter-cooperation among providers in '577 claim 1, it would have been obvious for one skill in the art at the time the invention was made to implement the plurality of entities in '577 so that information stored therein are details regarding the component specification and the service of manipulation data to achieve the 'desired

functionality' of a end users based on criteria to associate the sequence of components and the

various 'performance information' behind the stored details.

Claim Rejections - 35 USC § 101

4.   35 U.S.C. 101 reads as follows:

Whoever invents or discovers any new and useful process, machine, manufacture, or

composition of matter, or any new and useful improvement thereof, may obtain a patent

therefor, subject to the conditions and requirements of this title.

5.   Claims 29-34, 64-65, 67, 48, 59, 69 are rejected under 35 U.S.C. 101 because the claimed

invention is directed to non-statutory subject matter.

The current focus of the Patent Office in regard to statutory inventions under 35 U.S.C. § 101 for method claims and claims that
recite a judicial exception (software) is that the claimed invention recite a practical application. The following link on the World
Wide Web is the United States Patent And Trademark Office (USPTO) reference in terms of guidelines on a proper analysis on
35 U.S.C. §101 rejection.

http://www.uspto.gov/web/offices/pac/dapp/opla/preognotice/guidelines101_20051026.pdf

Specifically, **claims 29** recites 'apparatus" having processing systems for determining,

implementing code, causing server implementation, server to perform a service by interacting

and obtaining resultant sequence as code, and providing code for execution at the processing

system. The "using a processor" does not make it clear that processor is a hardware

device/apparatus integral to the "apparatus" or a mere software functionality or a user machine

provided external to the claimed "apparatus". As based from the Disclosure, all of these acts are

implemented by software or client/server processes, none of which explicitly equated to a

machine. The 'apparatus' claim as a whole amounts to mere listing of software-implemented

acts. In all, the 'system' or 'apparatus' amounts to listing of software functionality and this fall

into the "Functional Descriptive Material' deficiency as set forth in Annex IV (pg. 52-54) of the above 101 Guidelines; thus, the claim listing of mere software functionalities without explicit inclusion of hardware or tangible storage support cannot constitute a statutory category of subject matter. The 'apparatus' claim 29 is rejected as non-statutory, and dependent claims 30-34, 64-65, 67 are also rejected for not curing to the lack of hardware support deficiency required for the subject matter claimed to belong to a permissible category of subject matter, thereby constituting a non-statutory subject matter.

As per claim 48, the 'apparatus' as recited amounts to software-implemented act of determining, selecting, defining, and generating of specification; hence exhibits the same lack of hardware support as set forth above, whereas 'using a processor' does not make clear whether processor is hardware support being integral to the 'apparatus'. The 'apparatus' claim 48 is rejected as non-statutory for lack of hardware support required in order for the subject matter claimed to belong to a permissible category of subject matter, thereby constituting a non-statutory subject matter.

As per claim 59, this 'apparatus' claim recites a "processing system" for what appear (based on the Disclosure) to be mere software-implemented acts of providing, allowing, causing, interacting, obtaining, and as set forth above, the claim lacks hardware support to remedy to the 'functional descriptive material' (mere listing of software functionalities per se) deficiency as mentioned above. As processing system taken in a broad interpretation does not amount to a hardware device, the claim is rejected (as non-statutory) for not enable one to categorize the claim as any statutory subject matter.

Claim 69 also recites "apparatus" using a processor as set forth in claim 29, hence does not explicitly amount to including a hardware device provided within the claimed "apparatus"; and based on broad interpretation that 'input commands' are software or binary data, the claim as a whole does not constitute a statutory category of permissible subject matter.

Claim Rejections - 35 USC § 112

6.   The following is a quotation of the first paragraph of 35 U.S.C. 112:

> The specification shall contain a written description of the invention, and of the manner and process of making and using it, in such full, clear, concise, and exact terms as to enable any person skilled in the art to which it pertains, or with which it is most nearly connected, to make and use the same and shall set forth the best mode contemplated by the inventor of carrying out his invention.

7.   Claims 46-47 are rejected under 35 U.S.C. 112, first paragraph, as failing to comply with the written description requirement. The claim(s) contains subject matter which was not described in the specification in such a way as to reasonably convey to one skilled in the relevant art that the inventor(s), at the time the application was filed, had possession of the claimed invention.

As per claim 46, the 'determining a process of selecting one of the methods … received data' is nowhere disclosed without sufficient details to reasonably convey such "process" determining concept, based on the data received. According to the specifications, methods, data, component combination, service are determined; but nowhere is there a paragraph describing how a "process of selecting" methods is being determined by (emphasis added) any entity. One would be incapable of make or use the invention because of the lack of teaching and complete absence of a programmatic process of "determining" based on received data as claimed, the 'received data' having no structural relationship to any steps a) b) b-i) b-ii) of the claim. Nowhere in the Disclosure is there a clear depiction of a process of selecting being determined

responsive to "receiving data". Selecting a method in response to data would not be same as

determining a process for selecting, which entails <u>choosing a proper algorithm</u> to go about

selecting methods. The act of causing schematics to be built depending on the data contained

therein (see Applicant's Remarks pg. 19, top) cannot remotely be construed as 'determining a

process of selecting'. That is, the claim does not make it clear as to what action actually leads to

'received data' for this to reasonably convey a relationship to 'selecting one of the methods', or

under what antecedent basis (emphasis added) in the claim taken as a whole, any reception of

data has been achieved. Nowhere in the disclosure is there teaching that relates "reception of

data" with 'determining a process of selecting' as so recited for undue experimentation would

have to be used for one to implement the 'determining a process' as claimed. The 'determining a

process of selecting … methods' will be treated as selecting a method in a broad sense; i.e. this is

partially subsumed into the step recited as 'determining at least two methods of performing the

service'. Claim 46(including claim 47) for failing to provide enabling support for the above lack

of description will also be rejected as not fulfilling the requirement of a proper description.

Claim Rejections - 35 USC § 103

8.   The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all

obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

9.   Claims 1-19, 25-27, 29-34, 46-48, 51-54, 56-59, 62-69 are rejected under 35

U.S.C. 103(a) as being unpatentable over Rigole, USPN: 7,139,728 (herein Rigole) in view of

Lipkin, USPN: 6,721,747(herein Lipkin) and further in view of Gangopadhyay et al, USPN:

6,973,638 (herein Gangopadh).

As per claim 1, Rigole discloses a method of generating components, each component

being a representation of a respective data manipulation service provided by a component server,

the method including:

a) Determining a component combination, the component combination being a

combination of components representing a plurality of interconnected component servers for

performing a defined defining a series of data manipulations (col. 6 lines 36-46; service

Selection – col. 11; Bill module – col. 11-12, monitoring module – col. 12-13, best services

module – col. 13, search module - col.14, incentive module col. 14, Mining module, col. 14);

the component combination being defined in accordance with requirements (e.g. col. 7

line 50 to col. 8 line 9 - Note: participating computer as consumer having UI and form filling to

file a request - col. 5 lines 48-58 - to obtain fulfillment of portion of a collective/logical

chains/threads implemented by contributing provider's actions - col. 6 line 60 to col. 7 line 2 --

reads on combination of component – see one or more modules 2.03 - col. 10 lines 6-18 – each

component representing a participating service provided by a participating parties or provider of

service, including activity such as manipulating series of data via a GUI or scripting - see col. 6

lines 48-56; col. 7 line 12 to col. 8 line 23) to allow a user desired functionality to be achieved;

b) Implementing the component combination to generate the computer code by:

i) causing the implementation of a component server corresponding to each component in

the combination, each component server being implemented by a processing system; and,

ii) causing each component server to perform the respective data manipulation service in accordance with the defined series of data manipulations (e.g. col. 4 lines 44 to col. 5 line 17 – Note: consumer request based on implicating of a defined number of participating providers or parties – service sector - leading to communication of data – capture and processing, transmit - among the participating parties **reads on** a series of data manipulations – see col. 10 lines 39-48; Search, mining, pooling service and querying a database based on a received profile reads on data manipulation – see col. 14-15), at least some of the component servers performing the respective data manipulation_service by interacting with a data sequence (e.g. col. 6 lines 36-46; service Selection – col. 11; Bill module – col. 11-12, monitoring module – col. 12-13, best services module – col. 13, search module col.14, incentive module col. 14, Mining module, col. 14), at least some of the component servers performing the respective service by interacting with a data sequence (e.g. col. 5 lines 6-17; *data capture ... source ... another participating party's* computer ... presenting data to other computer - col. 5 line 64 to col. 6 line 18), and

at least some of the component servers performing the respective data manipulation service by at least one of:

interacting with one or more other component servers(e.g. col. 5 line 64 to col. 6 line 18; col. 5 lines 6-17; col. 6 lines 49-51; *streamline the interchange of data and files between ...* interchange party - col. 16 lines 42-57); and, causing the implementation of further components (Note: participating parties based on one single entry, in IPCS collaboration scheme using transfer of data among parties reads on further components – see logical chain - col. 6 line 60 to col. 7 line 2; single entry – col. 2 lines 58-64); and,

iii) Obtaining, as a result of the component servers performing the series of data

manipulations and from the data sequence, a resultant data sequence, the resultant data sequence

(col. 10 lines 49-67; col. 15 lines 66 to col. 16 line 21).

Rigole does not explicitly disclose implementing the component combination and data

manipulation to generate the computer code by implementing server and causing server to

perform a service; nor does Rigole explicitly disclose obtaining, as result from data

manipulations data sequence, a resultant data sequence being the computer code and providing

the computer code to a processing system, such that execution of the computer code by the

processing system causes the processing system to perform the desired functionality.

Consumers product being deliverables a market products or services or software assets

was a well-known concept at the time Rigole describes a collaboration among participating

parties to fulfill the consumer request for one of the variety of business or online products (see

Rigole: BACKGROUND: col. 1-2) where business applications are remotely stored as user's

programs (Users programs 2.03 – Fig. 2; col. 15 lines 47-52) within any participating computer

(consumer or provider) such that, for a user environment, code is necessary in the user's

operating on format being provided -- i.e. standardized languages (Java script, applet, HTML -

col. 8 lines 64 to col. 9 line 4) or in the user's context of dynamic script creating (appropriate

code from such languages… dynamically created … appropriate scripts – col. 7 lines 50-64).

Lipkin discloses business architecture with core services including offerings/catalog and

metadata-based managing tools (Fig. 7; col. 9 lines 47-67), providing XML-based interfaces (col.

23 lines 15-65; col. 87-88) for the business object author to implement business applications

(Fig. 4, 8a; col. 24), via development of message interconnect, management and finding/learning

(col. 5-8; Fig. 10) to generate APIs for query, delivery and integration (Fig. 14-16), via reuse of

bean API (col. 23-30; col. 33-34) from model diagrams (Fig. 8B-8C), code being embedded in

transmission script format in XML; e.g. to effectuate a context of a shopping cart(see col. 92-99;

col. 28 line 49 to col. 29 line 19).

Combining the interaction among graphical model to support business related XML

format transmission embedded with underlying bean code, code reuse via predefined Apis or

modules (with Java/bean) as in Lipkin, and reliance upon collaborating parties as in Rigole with

transmission of format for use with scripts operating in language as bean, javascript as above,

Gangopadh, in a collaboration architectural approach implicating multiple parties or

applications (col. 4 lines 46 to col. 5 lines 8) discloses generation of code based on events

representing at a user interface requiring asynchronous contribution of applications (Gangopadh:

Fig. 1A, 1B) to model or define a business process (Gangopadh: col. 3 lines 42-45) via activity

diagrams representation depicting interaction between the hierarchy of nodes (Gangopadh: Fig.

18) having connectors to other nodes, and where messages/arcs between business object nodes

represents a service request (Gangopadh: col. 10 lines 46-67) and where each request (or

message arc; Gangopadh: Fig. 11, 14-15) include implementation of a service port (Gangopadh:

col. 16 lines 17-25). Based on software being delivery as part of XML/metadata based service as

in Lipkin's and Rigole's provider matching programs to fulfill a user requests as set forth above,

it would have been obvious for one skill in the art at the time the invention was made to

implement the collaboration of servers or parties in Rigole, so that the resultant of data sequence

or series of manipulation thereof (performed by the service sectors or collaborating parties)

needed to implement the user request, would be resultant as to form a implemented (business

application as in Lipkin) code which when delivered to the users in the user endeavor's to develop a business application (as in Gangopadh multi-node services message with port definition, or Rigole's use of inter-communicating parties), the delivered code can execute at the user platform, i.e. execution of the computer code by the processing system causes the processing system to perform the desired business functionality.

As per claim 2, Rigole does not explicitly disclose at least some of the components including one or more ports for receiving and/or outputting data to be manipulated. However, each machine in Rigole's collaboration architecture represent a physical computer, hence a necessary presence of port identification at the physical layer of the ISO network hierarchy would be recognized in order for network protocol to be achieved (see Fig. 1 – TCP/IP ports) and based on which "data channels" are defined (see col. 10 lines 1-5) to carry transaction information. The use of port identification to represent one such service provider is disclosed in Gangopadh (col. 16 lines 17-25). Based on Rigole's definition of data channels between interrelated participating parties and based on the physical foundation constituting a network hierarchy, it would have been obvious for one skill in the art at the time the invention was made to implement the architecture of participating parties so that components related to service rendered by each parties (provider/consumer computers) is represented by a port (as in Gangopadh) with a listener for receiving asynchronously data into and out from the participating party's executing context (consumer or provider), because including a port for a specific component via a development GUI as in Gangopadh would enable a specific path for the user's application by which to expect data sent to and received from the provider, obviating otherwise extraneous packet filtering and port processing.

As per claim 3, Rigole does not explicitly disclose: each port having an agent adapted to control transfer of data to and from the component. Rigole discloses various format of data being communicated between the participating parties (col. 6 lines 47-59) with markup language protocol (HTML, XML, script) whereas a PDA client can be a thin client (col. 5 lines 41-53) and presents the IPCS capability as a intelligent agent to seek the best services for a request context(col. 3 lines 54-65). Following the scheme of using markup format from above, and the business application client-side development framework as in Gangopadh, Lipkin enhances the agent functionality in Rigole with a client communication within a framework architecture including a network of providers (Fig. 1) with use of dedicated agents to expedite or complement to the user's side capability with agent-specific functions (Lipkin: import agent, matching agent, delivery agent – Fig. 13, 15, 16). Based on Rigole's implementing of a service to capture XML forms (col. 5 bottom to col. 6 line 11) it would have been obvious for one skill in the art at the time the invention was made to implement the port identification as mentioned above (see claim 2) so that a listener associated thereto pertains to a agent or plurality thereof as in Lipkin, each with dedicated functionality to help process incoming data (parse XML data with matching its content for resolution) and redirecting data to the appropriate receiver, all of which to expedite communication process underlying the business endeavor level of the user, whereby enhance or complement to the client platform lack of resources (e.g. a thin PDA machine) at the application level.

As per claim 4, Rigole disclose having the component server (refer to claim 1 – Note: each participating parties or provider responsive to request from other clients to capture and render data in different form for returning response reads on component server):

receive data including a number of data portions (Fig. 1, 3); manipulate the data

sequence by

i) adding data portions into the sequence at a predetermined location;

ii) moving data portions from a first location to a second location within the sequence;

iii) removing data portions from the sequence; and, iv) modifying data portions in the sequence

(e.g. col. 5 lines 6-17; *data capture ... source ... another participating party's computer ...*

presenting data to other computer - col. 5 line 64 to col. 6 line 18; col. 5 line 64 to col. 6 line 18;

col. 6 lines 49-51).

As per claim 5, Rigole discloses using a processing system including a store, the

method including storing one or more of the data portions in the store (memory - Fig. 2; database

3.4 Fig. 3).

As per claim 6, Rigole discloses at least some of the components being formed from a

number of combined sub-components, the sub-components also being components (data capture,

presentation server - col. 5 line 64 to col. 6 line 21; administration module col. 7 line 3-18; Bill

module – col. 11-12, monitoring module – col. 12-13, best services module – col. 13, search

module - col.14, incentive module col. 14, Mining module, col. 14).

As per claim 7, Rigole discloses wherein at least some of the components servers being

data manipulations using at least one of: a) Manual manipulation of the data by an individual

(col. 6 lines 2-9 – Note: manipulation by a individual treated as software underlying the GUI

support to receive user input);

b) Computer code adapted to be executed by a processing system, to thereby manipulate of

the data automatically (col. 6 lines 35-47); and,

c) Combinations of sub-components, the sub-components also being components (refer to claim 6; *external interface module ... database interface module ... to be used by other modules* – col. 7 lines 19-45)

As per claim 8, Rigole discloses the method being performed using one or more processing systems (Fig. 1).

As per claim 9, Rigole disclose causing a first processing system to:

Select a number of components in response to input commands received from a user (refer to claim 7: manual manipulation; col. 9 line 14-25 – Note: consumer forms reads on user selecting of components provided in a XML form triggering service of mapping as per Best Services Selection module – see col. 13);

Define the component combination using the selected components (refer to claim 1; col. 7 line 65 to col. 8 line 9); and

Cause the component combination to be implemented such that the defined series of data manipulations is performed (refer to claim 1; col. 9 line 14-25).

As per claim 10, Rigole does not explicitly disclose at least some of the components including one or more ports, the method including causing the processing system to: a) Provide an indication of the ports of each selected component to the user; and, b) Interconnect selected ones of the ports in response to input commands from the user to thereby define the component combination. But the implementation of interconnected diagram/modules via message arc linking service nodes with user's specification of port has been disclosed in Gangopadh, as set forth in claim 1; hence the implementation of components via user interface as in Rigole to

provide ports for each components being selected, and to select ports via such user input would have been obvious in view of the rationale set forth in claim 2.

As per claims 11-12, Rigole discloses a second processing system to: Determine details of a number of components (col. 16 line 61 to col. 17 line 24); Provide at least an indication of the details to the user via the first processing system (Best Services Selection Module, col. 13; Comparison/Selection module col. 11; Services Search Module – col. 14);

causing the processing system to: Select respective ones of the components in response to input commands from the user (refer to claim 10); and, Provide the details of the selected components to the user via the first processing system (refer to claim 10).

As per claim 13, Rigole discloses second processing system to include: a) A store for storing the component specifications including at least one of:

An indication of the manipulation service (presentation server – col. 6 lines 12-21);

A graphical representation of the component (bitmap, graphics standard, WAV, JPEG, GIF - col. 6 lines 48-56); and,

and, b) A processor (Fig. 1), the method including causing the processor to: i) Obtain one or more component specifications from the store (Fig. 3; Selection Module – col. 11); and, ii) Provide the component specifications to the user via the first processing system (refer to claim 1; col. 16 line 61 to col. 17 line 20; Selection module - col. 11).

As per claim 14, Rigole does not explicitly disclose causing the first processing system to: a) Generate a graphical representation of the one or more selected components; and, b) Manipulate the graphical representation in response to input commands received from a user to thereby define the component combination.

But the endeavor to generate software component to be executed at the client machine

has been addressed in claim 1, where in light of Lipkin delivering of software from vendors,

including developing methodologies using CASE tools and activity state diagrams, with similar

teaching in Gangopadh's collaboration among applications, including a UI enabling the end user

(first processing system) to create tree of service nodes for representing the asynchronous

communication between collaborating applications, with inter-node messages/arcs implemented

as requests among the collaborating services, the event-based exchange and associated

messaging code defined via servicing ports, the representation in terms of a graphical model

view being subjected to user's manipulation (see Gangopadh: Fig. 3-5, 6, 10-18). The rationale

of obviousness is referred back to the rejection in claims 1, 2, 10.

As per claim 15, Rigole discloses said first processing system being coupled to one or

more component processing systems via a communications network, each component processing

system being adapted to implement one or more respective components servers, the method

including:

generating a service request for each component (col. 2 lines 58-64 – Note: user defining

a service sectors reads on request for each component performed per defined service - see col. 4

line 44 to col. 5 line 4; see: service Selection – col. 11; Bill module – col. 11-12, monitoring

module – col. 12-13, best services module – col. 13, search module , col. 14, incentive module

col. 14, Mining module, col. 14) in the component combination; and

transferring the service request to each component processing system via the

communications network, each component processing system being adapted to respond to

the service request to implement a component server for providing the data manipulation service

represented by the respective component (logical chain - col. 6 line 60 to col. 7 line 2; col. 5 lines 6-17; *data capture ... source ... another participating party's computer* ... presenting data to other computer - col. 5 line 64 to col. 6 line 18 – Note: a single profile to be accessible by plurality of participating providers reads on request to each said component processing system - see col. 4 lines 4-8).

As per claim 16, Rigole discloses: Determining any data required by the components; and, Providing the data in the service request (Fig 3; *Consumer ... data input ... parsed* – col. 9 lines 5-24).

As per claim 17, Gangopadh disclose each service request including an indication of the interconnections for each of the ports of the respective component (refer to claim 1) and the rationale for obviousness is referred back to claims 1-2.

As per claim 18, Rigole discloses causing each component processing system to:

a) implement one or more respective component in accordance with the received service request; and, b) Cause each component instance .server to: i) Interact with other components servers in accordance with the interconnections defined in the service request; and, ii) Perform any required data manipulations;

all of which being addressed in claim 1; e.g. using the participating parties and modules stored therein (Fig. 2), modules instantiated per the data capture server (col. 9 lines 5-24) based on the matching (Figure 3) leading to interaction among services (col. 6 line 60 to col. 7 line 2; col. 10 line 39-53)

As per claim 19, Rigole does not explicitly disclose causing each component processing system to: a) Implement a respective agent associated with each port; and, b) Cause each agent to

cooperate with an agent of another component in accordance with the defined interconnections,

to thereby allow data to be transferred between the ports.

The rationale as to use ports to define a respective service component subjected for a

service rendering has been addressed in claim 10; whereas associating an agent for each

component transaction via defined interconnected services or port identification thereof (based

on a UI graphical representation and code generation) has been addressed in claims 2-3. Based

on the transfer of data among participating parties in Rigole, it would have been obvious for one

skill in the art at the time the invention was made to implement data transfer so that ports is used

on conjunction with the respective agent as set forth in the above claims in order to associate port

specialization with supportive agents to obtain client platform improved performance and/or

communication benefits set forth in claim 3.

As per claim 25, Lipkin discloses transactional type of applications requiring

creating/implementing bean underlying a runtime context (col. 22 lines 53-61) with a bean

context (col. 23 lines 35-40) in terms of EJB model context embedded in metadata (col. 24 lines

49-60), while the rationale as to causing the Rigole's STN/IPCS system to generate code

destined for user runtime within the business application for the consumer to interpret script to

execute the desired functionality has been addressed as obvious in claim 1. Based on Lipkin's

bean context inside XML, script or metadata provided to users, it would have been obvious for

one skill in the art at the time the invention was made to Rigole business transaction and user's

runtime receiving script or web format so that code to execute the business functionality would

be dependent on context for the same reason as set forth in claim 1 as to why providing

generated code would be useful to the consumer's interpretation of the provided responses from

the STN/IPCS services.

As per claim 26, Rigole discloses code necessarily in various languages to support the

Web pages and script execution/creation at the consumer's runtime (col. 7 lines 50-64;

contextually indicated – col. 4 lines 37-44) where the runtime is based on context of the request

thence the result of data manipulation from participating parties (refer to claim 1; col. 4 lines 35-

44) hence has disclosed b) Perform the data manipulation service in accordance with the

determined context such that the performed data manipulation is dependent on the context; but

does not explicitly disclose a) Determine a context for the code; and,

However, the provision of manipulation service data response and associated bean for

script execution according to a context dependency has been addressed as obvious in claim 25,

and the rejection of the above limitation a) would incorporate the obviousness rationale

therefrom.

As per claim 27, Rigole discloses processing system (Fig. 1) including at least a

memory, stack and registers for executing the users' application which inherently necessitates a

runtime context including at least one of: a) The state of at least one of the registers, stack and

memory (col. 5 lines 30-41); the context including b) Other components in the defined

component combination (refer to defined combination of claim 1).

As per claim 29, Rigole discloses apparatus for generating computer code using

components, each component being representation of a respective data manipulation service

(refer to claim 1)provided by a component server, the apparatus including one or more

processing systems adapted to:

a) Determine, using a processor, a component combination, the component being a combination of components (refer to claim 1)representing a plurality of interconnected component servers for performing a defined series of data manipulations (refer to claim 1), end the component combination being defined in accordance with requirements to allow a user desired functionality to be achieved (refer to claim 1);

b) Implement, using the processor, the component combination to generate the computer code by:

i) causing the implementation of a component server corresponding to each component in the combination, each component server being implemented by a processing system(refer to claim 1); and

ii) causing each component server to perform the respective data manipulation service(refer to claim 1) in accordance with the defined series of data manipulations,

at least some of the component servers performing the respective data manipulation service by interacting with a data sequence(refer to claim 1), and at least some of the component servers performing the respective data manipulation service by at least one of:

(1) Interacting with one or more other component servers (refer to claim 1); and,

(2) Causing the implementation of further components (refer to claim 1); and,

iii) Obtaining as a result of the component servers performing the series of data manipulations and from the data sequence, a resultant data sequence, the resultant data sequence (refer to claim 1).

Rigole does not explicitly disclose implementing the component combination and data manipulation to generate the computer code by implementing component servers and causing the

servers to perform a service; nor does Rigole explicitly disclose obtaining, as result from data

manipulations data sequence, a resultant data sequence being the computer code and providing

the computer code to a processing system, such that execution of the computer code by the

processing system causes the processing system to perform the desired functionality.

However, these limitations have been rendered obvious as set forth in claim 1.

As per claim 30, Rigole discloses:

a) One or more component processing systems, each component processing system

being adapted to implement a respective component server (refer to claim 9; executed as a

separate logical server … physical device - col. 6 line 60 to col. 7 line 2; col. 10 line 39-53); and,

b) A first processing system (consumer systems 3 - Fig. 1), the first processing system

being adapted to:

i) Define the component combination in accordance with input commands received from

a user using an input device;

ii) Determine the component processing systems implementing the respective

components servers represented by the components of the component combination; and,

iii) Transfer service requests to each of the determined component processing systems

(for i, ii, iii, refer to claim 8, claim 1).

As per claim 31, Rigole discloses component processing system being adapted to: a)

Receive the service request; b) Generate a respective component instance; and, c) Perform the

data manipulation (refer to claims 14-15) service using the respective component instance.

As per claims 32-33, Rigole discloses a second processing system, the second processing

system being adapted to store details of available components(refer claims 11, 13); the second

processing system being adapted to obtain the details of a component from a respective

component processing system (refer to claims 12, 13, claim 18).

As per claim 34, Rigole discloses first processing system being adapted to cooperate

with the second processing system (Fig. 1) to thereby allow a user to: Select one or more of the

available components (refer to claim 12); and Define the component combination (refer to claim

1).

As per claim 46, Rigole discloses method of providing a dynamic component for

providing data manipulation services, the method including:

a). Determining a service to be performed(refer to claim 1);

b) Determining at least two methods of performing the service (service Selection – col.

11; Bill module – col. 11-12, monitoring module – col. 12-13, best services module – col. 13,

search module -col.14, incentive module col. 14, Mining module, col. 14 – Note: specialized

modules per participating service reads on predetermined one or two methods to perform data

manipulation), wherein the methods of performing the service utilizing respective components

include:

i) Selecting components to implement the desired services (refer to claim 1; Selection

module: *consumer ... select ... attributes ... define a best program* - col. 11), each component

being a representation of a respective data manipulation service provided by a component server

(Note: user activity such as manipulating series of data via a GUI or scripting - see col. 6 lines

48-56; col. 7 line 12 to col. 8 line 23- reads on component being representation of respective data

manipulation service – see Bill module – col. 11-12, monitoring module – col. 12-13, best

services module – col. 13, search module -col.14, incentive module col. 14, Mining module, col.

14 -- from another component performed by a providing and participating entity);

d) Generating a component specification defining a component data manipulation service

(col. 11 lines 20-55; Fig. 3; col. 15; col. 16 line 61 to col. 17 line 24; col. 6 lines 48-56; col. 7

line 12 to col. 8 line 23 – Note: participating entity receiving specification from a request and

communicating to other participating entity of same script format or GUI based profiling reads

on specification defining data manipulation service – see col. 6 lines 48-56; col. 7 line 12 to col.

8 line 23).

Rigole does not explicitly disclose: Defining a component schematic including at least:

A first schematic portion representing any common portion of each method of performing the

service; At least two second schematic portions representing any different portion of each

method of performing the services; nor does Rigole explicitly disclose selector agent for

selecting a respective one of the second schematic portions.

The implication for code to be developed or provided as necessary embedded executables

(Java API, applet, Javascript) to support the use of the transmitted format via 'data channels' in

terms of page interpretation or dynamic script creation in Rigole consumers side (see Rigole: col.

8 lines 64 to col. 9 line 4) after receiving such response format from the participating parties

(server response) has been addressed in claim 1. And developing of code has been taught in

Lipkin BDK tool server to enable metadata (Lipkin: Fig. 15-16) to derive widget-model (view

892, Fig. 8B; Fig. 8C) graphically in order to assemble bean code as to help the network of users

to develop their desired business management application using the underlying code embedded

in the transmitted XML forms (see col. 28 line 40 to col. 29 line 59), whereas development of

code in form of graphical model (a top level reaching out to lower levels – top common

component branching to at least two lower components). Following Lipkin, the use of

interconnected diagram representation (or activity diagrams) as a in a schematic form is

disclosed by Gangopadh (see Fig. 2-18) to implement collaboration among components of the

viewed model where methods invocation interacting with other components (Gangopadh: col. 12

line 20 to col. 14 line 15; col. 4 line 63 to col. 5 ine 24) on the interconnected diagram

representation (top common nodes branching to two children nodes) are selected and

implemented by the developers. Based on the use of code to support a transmitted form of

stream whereby to request/specify service invocation or to obtain data from collaborating

services (participating parties in Rigole based on consumers' selection), it would have been

obvious for one of ordinary skill in the art to implement the service providers in Rigole based on

Lipkin's teaching metadata-derived model view, so that Java-APIs type of code (in Rigole or

Lipkin) generated based on such model includes schematic representation (i.e. any common

portion of method service and at least two second schematic portions representing any different

portion of each method of performing the service) being generated for one to choose a form of

invocation requiring data between entities on the schematic representation as taught in

Gangopadh, whereby at least one or two methods or function calls to obtain respective data of

the graph nodes are implemented as code to support the execution of page/scripts being

instrumental in the consumer's runtime in Rigole and in Lipkin. That is, the motivation as to

implement code with API method calls (at least two to create a request/receipt of data) as set

forth above would include the same benefits as presented in the rational of claim 1.

Nor does Rigole explicitly disclose upon receiving data, determining a process (refer to the USC 112 Rejection; i.e. determining process of selecting treated as selectively determining methods or techniques or APIs) of selecting one of the methods in accordance with the received data. Determining methods or techniques or program implementation functions or interfaces is disclosed as Bean/Applet API to support realization of service or methods being determined in Rigole's GUI-based approach, i.e. implementation based on specifications parsed from communications among participating parties. Accordingly, the choosing of methods for Java or Bean/applet API falls under the ambit of generating code using a viewable interconnect representation (schematic component) being used by developers to specify code calling between node/blocks of the schematic representation as set forth above using Lipkin, and Gangopadh; hence the rationale for rendering the above obvious would be same as above.

As per claim 47, Rigole does not explicitly disclose defining an agent associated with each input or output, the agent being adapted to cooperate with an agent of another component in accordance with the defined interconnections, to thereby allow data to be transferred between the ports of the components. But the expediting process by which agents are allocated per ports to process in and out data from Rigole communication paradigm has been addressed in claims 2-3.

As per claim 48, Rigole (in view of Lipkin, Gangopadh) discloses apparatus for providing a dynamic component for providing data manipulation services, the apparatus including a processing system for:

a) Determining, using a processor, a service to be performed;

b) Determining, using the processor, at least two methods of performing the service, wherein the methods of performing the service utilizing respective components included:

i) Selecting components to implement the desired services each component being a representation of a respective data manipulation service provided by a component server;

ii) Defining a component schematic including at least: (1) A first schematic portion representing any common portion of each method of performing the service; (2) At least two second schematic portions representing any different portion of each method of performing the services; and, (3) A selector agent for selecting a respective one of the second schematic portions;

c) Upon receiving data, determining, using the processor, a method of selecting one of the methods in accordance with received data; and,

d) Generating a component specification defining a component representing the data manipulation service;

all of which having been addressed in claim 46.

As per claim 51, this claim recites the same subject matter of claim 1; hence would incorporate the rejection as set forth therein.

As per claim 52, Rigole (in view of Lipkin, Gangopadh) discloses: Allowing users to select components; and, b) Providing users with a component specification for each selected component, each component specification defining the data manipulation service (refer to claim 9) and port specifications defining data to be received at or output from respective ports (refer to claim 10).

As per claim 53, Rigole discloses obtaining the component specification for a respective component from component processing system implementing the component server represented by the component (e.g. streamline the interchange of data and files between … standardized …

pages … input forms - col. 16 lines 42-57 - Note: implementation or formatted page/ structured Web recommendations based on specifications parsed from inter-communicated requests among participating parties or interchange party reads on obtaining specifications from a system implementing a component server – see links to other relevant sites - col. 16 line 61 to col. 17 line 20; Selection module - col. 11).

       As per claim 54, Rigole discloses one or more processing systems coupled to a user end station via a communications network, the method including: a) Allowing the user to select the components using the end station; and, b) Transferring the component specifications to the end station from one or more of the processing systems (refer to claim 1, 8-9, 15).

       As per claim 56, Rigole discloses: a) Causing the end station to generate service requests in accordance with the component combination; and, b) Transferring the service request to entity processing systems thereby causing the entity processing systems to perform the data manipulation defined by the component (refer to claims 1, 8-9, 15, 17).

       As per claim 57, the limitation as to the component combination defining connections between the components servers, the service requests including connection information determined by the end station from the component specifications falls under the ambit of port specifications and interconnections of component combination associating with the services to be performed by one or more entities; and this has been addressed in claims 2-3, 10.

       As per claim 58, Rigole discloses:

       causing the component processing systems to: a) Generate one or more component servers in accordance with the received service request; b) Cause each component server to:

i) Cooperate with other components-servers to send and/or receive data; and ii) Perform
the required data manipulation service;

all of which having been addressed in claims 4, 6-7, 15, 18)

As per claim 59, Rigole (in combination with Lipkin, Gangopadh) discloses apparatus
for generating computer code by allowing users to manipulate a data sequence, the apparatus
including a processing system for:

a) Providing details of a number of components, each component being a representation
of a respective data manipulation service provided by a component server;

b) Allowing users to define a component combination, the component combination
being a combination of components and representing a plurality of interconnected component
servers for performing a defined defining a series of data manipulation services;

c) implementing, using a processor implement the component combination to generate
the computer code by: i) Causing the implementation of a component server corresponding to
each component in the combination, each component server being implemented by a processing
system; ii) Causing each component server to perform the respective data manipulation service
in accordance with the defined series of data manipulations,

at least some of the component servers performing the respective data manipulation
service by interacting with a data sequence, and at least some of the component servers
performing the respective data manipulation service by at least one of:

(1) Interacting with one or more other component servers; and,

(2) Causing the implementation of further components; and,

iii) Obtaining as a result of the component servers performing the series of data

manipulations and from the data sequence, a resultant data sequence, the resultant data sequence

being the computer Code; and,

d) Providing the computer code to a processing system, such that execution of the

computer code by the processing system causes the processing system to perform the desired

functionality;

all of which having been addressed in claim 1.

As per claim 62, Rigole discloses implementing at least some component servers (refer

to claim 29) by: a) Determining a context from the defined component combination; and, b)

Performing the data manipulation service in accordance with the determined context(context –

col. 4 lines 33-44)

As per claim 63, Rigole discloses wherein at least some of the components include a

number of predetermined techniques for performing the respective data manipulation service

(service Selection – col. 11; Bill module – col. 11-12, monitoring module – col. 12-13, best

services module – col. 13, search module -col.14, incentive module col. 14, Mining module, col.

14 – Note: specialized modules per participating service reads on predetermined techniques to

perform data manipulation), and

wherein the method includes implementing at least some of the component servers by:

Selecting one of the predetermined techniques based on the component combination

(refer to claim 1, 29; service Selection – col. 11); and,

Performing the respective data manipulation service using with the selected

predetermined technique (col. 5 lines 6-17; *data capture ... source ... another participating*

party's computer … presenting data to other computer - col. 5 line 64 to col. 6 line 18 - Note:

transferring of streams of data among servers based on the state of the data received and the next

function performed by the next servers reads on implementing servers for handling data

manipulation using techniques selected via interaction among servers)

> **As per claims 64-65**, refer to claims 62-63.

> **As per claim 66**, Rigole does not explicitly disclose wherein the data sequence is a

binary file including bytes, and wherein at least some of the component servers perform the

respective data manipulation services by manipulating the bytes of the binary file, and wherein

the generated computer code is executable binary code.

> But receiving bytecodes as embedded API inside Web content or pages for manipulation

is indicated in Rigole's approach using SGML format (see col. 6 lines 47-56; Javascript, Java -

col. 7 lines 47-51) where the browser-based consumer receiving a service response can

manipulate the binary file or Java byte format embedded in HTML page or scripting form (see

col. 8 line 42- to col. 9 line 4). Based on the rationale that deliverable code can be executable

code or bytes as set forth in claim 1 using Lipkin and Gangopdh, the limitation as to including

binary file for a component servers or participating party to manipulate data via Rigole's

scripting or page interpretation based on HTML, SGML or messaging approach would have been

obvious, because streaming data via this protocol including API or embedded executable therein

to support provisioning services as endeavored in Rigole, Lipkin and Gangopadh would benefit

from the methodologies of the web and its inherent Java core, as well as the extended aspect of

multi-platform W3C protocols; e.g. scripting with embedded Java APIs for target computer

manipulation.

As per claim 67, refer to the rationale of claim 66.

As per claim 68, Rigole discloses a method of generating computer code using components, each component being a representation of a respective data manipulation service provided by a component server, the method including:

a) determining a component combination, the component combination being a combination of components representing a plurality of interconnected component servers (refer to claim 1) for performing a defined series of data manipulations, the component combination being defined in accordance with requirements to allow a user desired functionality to be achieved (refer to claim 1), the component combination being determined by:

i) providing input commands from the user to a first processing system (refer to claim 9); ii) Based on the input commands, obtaining a plurality of component specifications each including an indication of the data manipulation service represented by a respective component (refer to claims 2, 3 10),

at least some of the plurality of components including inputs and outputs (refer to claims 2, 3) for receiving and/or outputting data to be manipulated (see below); and,

iii) using the first processing system to define the component combination by interconnecting selected ones of the plurality of components (refer to claim 1) by interconnecting inputs and outputs of certain of the selected ones of the plurality of components in response to input commands received from the user (refer to the rationale in claims 9-10);

b) Implementing the component combination to generate the computer code by:

i) Causing the implementation of a component server corresponding to each component

in the combination (refer to claim 1), each component server being implemented by a processing

system; and,

ii) Causing each component server to perform the respective data manipulation service

(refer to claim 1) in accordance with the defined series of data manipulations, at least some of the

component servers performing the respective data manipulation service by interacting with a data

sequence (refer to claim 1), and at least some of the component servers performing the respective

data manipulation service by at least one of:

(1) Interacting with one or more other component servers (refer to claim 1); and, (2)

Causing the implementation of further components (refer to claim 1); and,

iii) Obtaining, as a result of the component servers performing the series of data

manipulations and from the data sequence, a resultant data sequence, the resultant data sequence

being the computer code(refer to rationale in claim 1); and,

c) Providing the computer code to a processing system, such that execution of the

computer code by the processing system causes the processing system to perform the desired

functionality (refer to rationale in claim 1).

As per claim 69, Rigole discloses (in view of Lipkin, Gangopadh) apparatus for

generating computer code using components, each component being a representation of a

respective data manipulation service provided by a component server, the apparatus, including

one or more processing systems adapted to:

a) Determine, using a processor, a component combination, the component being a

combination of components embodying a respective data manipulation service and representing

a plurality of interconnected component servers for performing a defined series of data

manipulations, the component combination being, defined in accordance with requirements to

allow a user desired functionality to be achieved,

the component combination being determined by:

i) Providing input commands from the user to a first processing system;

ii) Based on the input commands, obtaining a plurality of component specifications each

including an indication of the data manipulation service represented by a respective component,

at least some of the plurality of components including inputs and outputs for receiving and/or

outputting data to be manipulated; and,

iii) Using the first processing system to define the component combination by

interconnecting :selected ones of the plurality of components by interconnecting inputs and

outputs of certain of the selected ones of the plurality of components in response to input

commands received from the user;

b) Implement, using the processor, the component combination to generate the computer

code by:

i) Causing the implementation of a component server corresponding to each component

in the combination, each component server being implemented by a processing system; and,

ii) Causing each component server to perform the respective data manipulation service

in accordance with the defined series of data manipulations, at least some of the component

servers performing the respective data manipulation service by interacting with a data sequence,

and at least some of the component servers performing the respective data manipulation service

by at least one of:

(1) Interacting with one or more other component servers; and, (2) .Causing the

implementation of further components; and,

iii) Obtaining as a result of the component servers performing the series of data

manipulations and from the data sequence, a resultant data sequence, the resultant data sequence

being the computer code; and

c) Providing the computer code to a processing system, such that execution of the

computer code by the processing system causes the processing system to perform the desired

functionality;

all of which having been addressed in claim 68.

10.   Claims 20-24, 37-43, 55 are rejected under 35 U.S.C. 103(a) as being unpatentable over

Rigole, USPN: 7,139,728 (herein Rigole) in view of Lipkin, USPN: 6,721,747 and

Gangopadhyay et al, USPN: 6,973,638; further in view of Hanagan et al, USPubN:

2001/0056362 (herein Hanagan)

As per claims 20-21, Rigole does not explicitly disclose causing, the second processing

system to:

a) Determine performance information, the performance information being representative

of one or more criteria regarding the implementation of the components servers;

b) Provide the performance information to a user, the user selecting the components in

accordance with the performance information, the performance information including **at least**

one of:

a) an indication of the component processing implementing the component server; b) an

indication of the geographical location of the component processing system; c) an indication of

the duration for implementing the component server; d) an indication of a cost associated with implementing the respective component server; and, e) A rating, the rating being indicative of the success of the component server.

Rigole discloses a provider service to electronically notify the user/consumer with information regarding billing, price, rating (electronic copies, notifies by email, web pages– col. 12 lines 3-42; col. 10, lines 39-45; price, rating, particular region – col. 11 lines 38-51) or present web pages to the accessing client in order for client/user to review details or payment options about the billing before payment; hence the or performance information/attributes for user to select/weigh is disclosed or would have been obvious. Regarding the latter, the same business practice by which a provider provides information about tariff or pricing regarding the service or option presented to the user is disclosed in Hanagan (see para 0196, pg. 12) where a customer care service prepare information for a price plan (para 0185, pg. 11) with disclosing of charges (para 0188, pg. 11), bill information, information associated with transmission speed as well as template presenting type of services are also provided to the client (para 0189-0190 pg. 11), i.e. performance indication. Based on Rigole's purpose to reach effective shopping cRigolebility and extensibility to various group of consumers (see Rigole: col. 1-2) it would have been obvious for one skill in the art at the time the invention was made to implement IPCS module by Rigole in terms of conveying billing or offering/rating information as set forth above so that performance indication regarding cost of services and speed of transmission be made available to the prospect client for all necessary charges and options be considered, thereby compacting the effectiveness of the transaction in view of obtaining satisfaction or expediting a reach towards user agreement based on the amount of information delivered prior to actually

providing the service, as endeavored in Rigole and Hanagan; e.g. providing a rating as well as price or speed type performance for the user to weigh in order attract a wider population of customer leading to a larger geographical groups.

As per claim 22, Rigole discloses providing a number of different components that are representations of data manipulation services provided by different component servers, the different components servers being implemented by component processing systems (refer to claim 1 – Note: data specified by user's GUI in view of the intended service to be represented as a profile or request for a service to be sent to other participating providers reads on representation of services provided by component servers) but does not explicitly disclose:

a) Providing a number of such components as representations of substantially equivalent data manipulation services provided by different component servers; and

b) Inducing competition between providers of the component processing system to thereby drive improvement of the components servers.

The concept of allowing user to shop or choose among service sectors entails a inducing a form of action reaching improvement of quality by each sectors when rating, sale terms information is divulged to the consumers as well as pricing and available services information (see col. 11 lines 20-55). Since Rigole's network provides various services offered over many sectors, which can be selected by consumers pertaining within extended geographic locations, a setting for competition among sectors or vendors is suggested. Hanagan discloses competition in prices for a same product that has to consider customer's choice (para 0009, pg. 1; para 0045, pg. 2) and that customer care, billing robust performance are essential for competing in a market (para 0084, pg. 4). It would have been obvious for one skill in the art at the time the invention

was made to implement the competition-oriented distribution of sectors (the better the

competition - col. 4 line 50 to col. 5 line 4) and the provision of billing, rating, available sectors

information to the prospected buyers/consumer as in Rigole, based on the teachings by Hanagan

so that the information is underlying a purport to induce, foster or promote competition among

equivalent services or product vendors in terms of price, rating, billing conditions, terms of

offerings and performance (see Hanagan) as set forth above, because competition is fundamental

in today's market for it is underlines the ultimate goal to as fulfilling satisfaction of the consumer

(see Rigole: the more convenient the STN will be for the consumers - col. 4 lines 52-57)

As per claims 23-24, Rigole does not explicitly disclose including generating revenue by

charging a cost for the use of each component; including:

a) Providing at least some of the revenue to a respective provider of the component

processing system implementing the component server represented by the component; and,.

b) Having the operator of the second processing system retain at least some of the

revenue.

The notion of provider of business and assets to client via services rendered as in Rigole

entails generating revenue or profit to the organization behind the services as this profit

generating was a well-known concept. Human operator or vendor of the services rendered being

able to receive the profit was inherent to the above concept; and this would be construed as an

official notice. Based on the payment module to consolidate the profit behind Rigole's IPCS

network to support payment from client being charged of the services rendered (see Rigole: col.

11 lines 57 to col. 12 line 42), it would have been obvious for one skill in the art at the time the

invention was made to implement Rigole's collaborative service framework so as to be able to

implement such profit by generating of revenue so that revenue can be selectively distributed

among service renderers or operators of each server processing system (e.g. service can retain

partial or whole revenue), because, as this is a well-known concept, revenue should be a reward

only based on performance and contributing factor for each service as distributed in Rigole's

IPCS network..

As per claims 37-38, Rigole discloses method of allowing users to manipulate a data

sequence, the method including using one or more processing systems coupled to a number of

end stations via a communications network, using one or more processing systems to:

a) Store details of a number of components, each component being a representation of a

respective data manipulation service (refer to claims 11-13) provided by a respective component

server entity (refer to claim 1) and the details being at least partly based on a component

specification from a respective processing system (refer to claim 1) for implementing the

component server(refer to claim 9);

b) Provide details of selected components to users, thereby allowing the users to select

components (see claim 12; col. 11 lines 6-55) and define a component combination defining a

series of data manipulation services for manipulating the data sequence using an end station

(refer to claim 1, claim 9); the component combination being a combination of components and

representing a plurality of interconnected component servers (refer to claim 1) for performing the

defined series of data manipulation services.

Rigole does not explicitly disclose: determine performance information representative of

one or more criteria regarding the implementation of the components; and, provide the

performance information to a user, the user selecting the components in accordance with the

performance information;

the performance information including at least one of: a) An indication of the component

processing system implementing the component; b) An indication of the geographical location of

the entity; c) An indication of the duration for implementing the component; d) An indication of a

cost associated with implementing the respective component; and, e) A rating, the rating being

indicative of the success of the component.

But the information provided to users serving a weigh or criteria towards quality,

offerings, cost and performance to enable review and consideration by the customers prior to

deciding on a service or a product purchase has been taught in Rigole (see claim 20), whereas the

performance information (cost, rating, duration, location) in regard to support the customer's

endeavor to go about the purchase has been deemed <u>disclosed (by Rigole) OR obvious</u> as set

forth in claim 20-21 from above (in view of Rigole/Hanagan).

As per claim 39, refer to claim 22.

As per claims 40-41, refer to claims 23-24

As per claim 43, Rigole discloses an apparatus for allowing users to manipulate a data

sequence, the apparatus including one or more processing systems coupled to a number of end

stations via a communications network (refer to claim 37), the one or more processing stations

adapted to:

a) Store details of a number of components, each component being a representation of

representing a respective data manipulation service (refer to claim 37) provided by a respective

component server (refer to claim 37) and the details being at least partly based on a component

specification from a respective component processing system for implementing the component

server (see col. 6 lines 48-56; col. 7 line 12 to col. 8 line 23);

b) Provide details of selected components to users, thereby allowing the users to .select

components and define a component combination defining a series of data manipulation

services(refer to claim 37) for manipulating the data sequence using an end station, the

component: combination being a combination of components and representing a plurality of

interconnected component servers for performing the defined series of data manipulation

services (refer to claim 37; col. 6 lines 48-56; col. 7 line 12 to col. 8 line 23);

c) Determine performance information representative of one or more criteria regarding

the implementation of the components; and d) Provide the performance information to a user, the

user selecting the components in accordance with the performance information (refer to rationale

in claim 20).

As per claim 55, refer to the rationale set forth for claim 21.

11.   Claim 28 is rejected under 35 U.S.C. 103(a) as being unpatentable over Rigole, USPN:

7,139,728 (herein Rigole) in view of Lipkin, USPN: 6, 721, 747 (herein Lipkin) and further in

view of Gangopadhyay et al, USPN: 6,973,638; further in view of Lapstun et al, USPN:

7,154,638 (herein Lapstun)

As per claim 28, Rigole does not explicitly disclose including making the data viewing

(col. 6 lines 12-21, 47-55) and printing by a user (col. 10 lines 20-32) at the end of the

transmission channels supporting the services within the STN system. The embedding of

manipulation context dependent by at least one of: a) Dithering; b) Meshing; and, c) Obscuring.

However, Rigole discloses printing in a context of presentation server transmitting format for

specifications inside W3c markup language as in Rigole or Lipkin has been well known for

presenting data or for enabling rendering within a page. Lapstun discloses server responding to

request form (Fig. 1) using tag encoding methodology (W3C) where netpages include formatted

elements as hyperlink or video/graphic object (Fig. 27-30) with form element for rendering (Fig.

32, Fig. 38a) in conjunction a controller operating with printed page encoder (Fig. 44-46, 47-52),

the encoding used in conjunction with a print engine adjusted to tone/ink (i.e. lightening or

obscuring) parameters and print head including dithering during the layout of pipeline of streams

(col. 49 lines 5-26). It would have been obvious for one skill in the art at the time the invention

was made to implement the markup format transmission via data channels by Rigole system so

that the service context for data manipulation would include presentation context regarding

printable format as set forth above, including server side tag encoding with consideration for

obscuring and dithering as taught in Lapstun, because markup tag embedding via standardized,

platform independent format/language, as shown in Lapstun or in Rigole/Lipkin, was well-

known at the time the invention was made, and would support stream communication between

servers and consumers, such that web pages rendering destined for the consumers receiving

manipulation data response from the STN/IPCS, can provide proper encoding for printing

purposes as contemplated in Rigole.

Response to Arguments

12.   Applicant's arguments filed 11/22/10 have been fully considered but they are not

persuasive or rather moot. Following are the Examiner's observation in regard thereto.

(A)   Applicants have submitted that the "service" of the present invention is distinct from that

disclosed in Rigole, in that component combination, as a representation of data manipulation

which when implemented as component server cause series of manipulation based on the defined

functionality which results in computer code, unlike services selected by the user as in Rigole

(Applicant's Remarks pg. 20, middle). The added language to the claim has been interpreted and

matched with proper teachings by Rigole; and one cannot see where novelty lies or non-

obviousness of the above features is established when the Argument is founded on mere

paraphrasing an alleged "patentable" (newly introduced) subject matter (as opposed to pointing

out where a exact language would be distinguishing from a cited part in Rigole) that appears to

have no direct mapping to or connection with the cited portions of the reference. That is, the

cited portions in Rigole apply **either** to a previous claim language **or** the current amendment,

hence the paraphrased subject matter as above cannot fall into either claim language, for the

above paraphrasing and associated allegation would not be commensurate with the exact Office

action that responds to a very specific claimed invention of a give/precise time context.

Applicant's arguments fail to comply with 37 CFR 1.111(b) because they amount to a general

allegation that the claims define a patentable invention without specifically pointing out how the

language of the claims patentably distinguishes them from the reference.

(B)   Applicants have submitted that on skill in the art would not found it obvious to gear

Rigole in generating computer code because Rigole in fact does not discloes 'a representation of

a respective data manipulation service … by a component server … for performing a defined

series of data manipulations' (Applicant's Remarks pg. 21) A defined series has been construed

as embedded content inside a script or manipulation at a browser GUI to formulate a request or

to modify content in a page to generate a modified profile or message destined to a requester or

to another participating party; and to this effect, Rigole has disclosed "defined series of data

manipulation'. Besides, the claim language in question is a added to the previous subject matter

under prosecution; and any argument related to an amendment cannot be deemed appropriate a

prima facie case of rebut; that is, the argument considered largely moot in view of the current

Office Action.

(C)   Applicants have submitted that apart from Rigole's complete non-disclosing of the

claimed elements in claim 1, the combination using Lipkin and Gangopadh failed to render

obvious the independent claims 1, 29, 59; and accordingly claims 37 and 43 cannot be disclosed

by the combination of references in regard to 'each component being a representation of a

respective data manipulation service ... the defined series of data manipulation services'

(Applicant's Remarks pg. 24). The argument is a rehash of that expressed earlier and considered

largely moot in light of the response set forth in sections A or B.

     In all, the arguments rely on the added language to proffer patentability of the added

features (e.g. Applicant's Remarks pg. 24-25); and this is insufficient to overcome the grounds of

rejection that have been necessitated by the Amendments.

Conclusion

13.   **THIS ACTION IS MADE FINAL.** Applicant is reminded of the extension of time

policy as set forth in 37 CFR 1.136(a).

     A shortened statutory period for reply to this final action is set to expire THREE

MONTHS from the mailing date of this action. In the event a first reply is filed within TWO

MONTHS of the mailing date of this final action and the advisory action is not mailed until after

the end of the THREE-MONTH shortened statutory period, then the shortened statutory period

will expire on the date the advisory action is mailed, and any extension fee pursuant to 37

CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event,

however, will the statutory period for reply expire later than SIX MONTHS from the mailing

date of this final action.

Any inquiry concerning this communication or earlier communications from the

examiner should be directed to Tuan A Vu whose telephone number is (571) 272-3735. The

examiner can normally be reached on 8AM-4:30PM/Mon-Fri.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's

supervisor, Lewis Bullock can be reached on (571)272-3759.

The fax phone number for the organization where this application or proceeding is

assigned is (571) 273-3735 (for non-official correspondence - please consult Examiner before

using) or 571-273-8300 (for official correspondence) or redirected to customer service at 571-

272-3609.

Any inquiry of a general nature or relating to the status of this application should be

directed to the TC 2100 Group receptionist: 571-272-2100.

Information regarding the status of an application may be obtained from the Patent Application

Information Retrieval (PAIR) system. Status information for published applications may be

obtained from either Private PAIR or Public PAIR. Status information for unpublished

applications is available through Private PAIR only. For more information about the PAIR

system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR

system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

/Tuan A Vu/

Primary Examiner, Art Unit 2193

January 21, 2011